



US007376147B2

(12) **United States Patent**  
**Seto et al.**

(10) **Patent No.:** **US 7,376,147 B2**  
(45) **Date of Patent:** **May 20, 2008**

- (54) **ADAPTOR SUPPORTING DIFFERENT PROTOCOLS**
- (75) Inventors: **Pak-Lung Seto**, Shrewsbury, MA (US);  
**Deif N. Atallah**, Chandler, MA (US)
- (73) Assignee: **Intel Corporation**, Santa Clara, CA (US)

6,906,918 B2	6/2005	Rabinovitz	
7,093,033 B2 *	8/2006	Beckett et al. ....	710/11
2002/0083120 A1 *	6/2002	Soltis .....	709/200
2002/0124108 A1 *	9/2002	Terrell et al. ....	709/245
2003/0033477 A1	2/2003	Johnson et al.	
2003/0065686 A1 *	4/2003	Callahan et al. ....	707/200

(Continued)

- (\* ) Notice: Subject to any disclaimer, the term of this patent is extended or adjusted under 35 U.S.C. 154(b) by 834 days.

FOREIGN PATENT DOCUMENTS

EP	1363429	11/2003
----	---------	---------

(Continued)

- (21) Appl. No.: **10/742,029**

- (22) Filed: **Dec. 18, 2003**

OTHER PUBLICATIONS

- (65) **Prior Publication Data**  
US 2005/0138191 A1 Jun. 23, 2005

PCT/US2004/040138 International Search Report & Written Opinion mailed Sep. 23, 2005.

(Continued)

- (51) **Int. Cl.**  
**H04J 3/16** (2006.01)  
**H04J 3/22** (2006.01)

*Primary Examiner*—Edan Orgad  
*Assistant Examiner*—Chuong Ho

- (52) **U.S. Cl.** ..... **370/465**; 370/466; 710/104
- (58) **Field of Classification Search** ..... 370/465, 370/466; 710/11, 14, 15, 104, 100, 105, 710/305; 711/170, 100; 709/230

(74) *Attorney, Agent, or Firm*—Konrad Raynes & Victor LLP; David W. Victor

See application file for complete search history.

(57) **ABSTRACT**

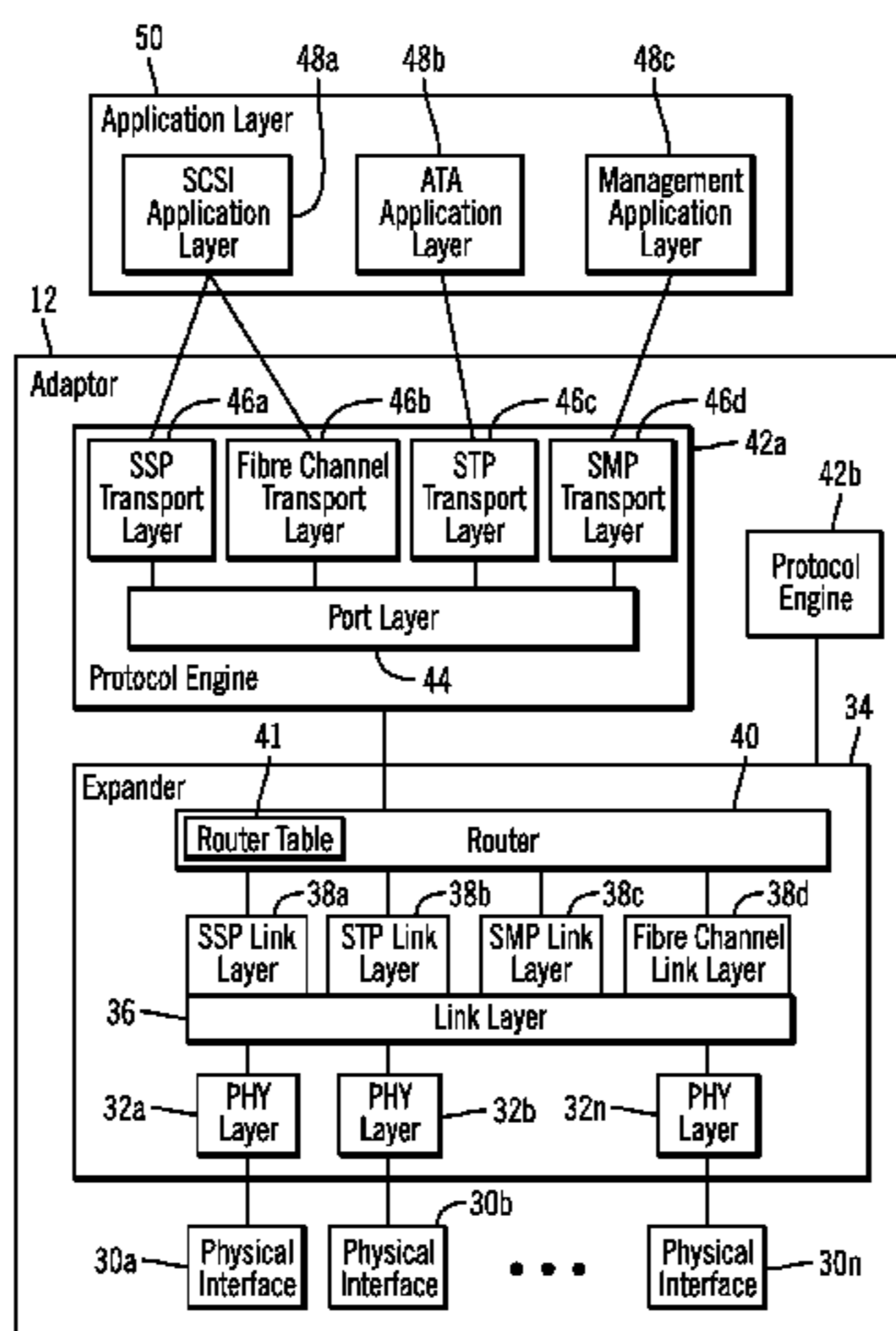
- (56) **References Cited**

U.S. PATENT DOCUMENTS

- |              |         |                   |
|--------------|---------|-------------------|
| 5,237,567 A  | 8/1993  | Nay et al.        |
| 5,692,128 A  | 11/1997 | Bolles et al.     |
| 5,706,440 A  | 1/1998  | Compliment et al. |
| 6,038,400 A  | 3/2000  | Bell et al.       |
| 6,044,411 A  | 3/2000  | Berglund et al.   |
| 6,289,405 B1 | 9/2001  | Movall et al.     |
| 6,333,940 B1 | 12/2001 | Baydar et al.     |
| 6,351,375 B1 | 2/2002  | Hsieh et al.      |
| 6,438,535 B1 | 8/2002  | Benjamin et al.   |
| 6,532,547 B1 | 3/2003  | Wilcox            |
| 6,553,005 B1 | 4/2003  | Skirmont et al.   |
| 6,853,546 B2 | 2/2005  | Rabinovitz        |
| 6,856,508 B2 | 2/2005  | Rabinovitz        |

Provided are a method, adaptor, system, and program for receiving a transmission at one of multiple connections. Information is maintained on storage interconnect architectures and transmission characteristics, wherein the storage interconnect architectures have different transmission characteristics. At least one transmission characteristic of the received transmission is determined and a determination is made from the information of the storage interconnect architecture associated with the determined transmission characteristic. The information on the determined storage interconnect architecture is used to process the transmission and determine a transport layer for the received transmission, wherein there is one transport layer for each supported transport protocol. The transmission is forwarded to the determined transport layer.

**36 Claims, 5 Drawing Sheets**



U.S. PATENT DOCUMENTS

2003/0184902 A1 10/2003 Thiesfeld  
 2003/0193776 A1 10/2003 Bicknell et al.  
 2003/0221061 A1\* 11/2003 El-Batal et al. .... 711/114  
 2004/0010612 A1 1/2004 Pandya  
 2004/0133570 A1\* 7/2004 Soltis ..... 707/3  
 2005/0251588 A1 11/2005 Hoch et al.  
 2007/0067537 A1 3/2007 Seto

FOREIGN PATENT DOCUMENTS

GB 2354137 3/2001

OTHER PUBLICATIONS

ANSI INCITS 376-2003: Serial Attached SCSI (SAS).  
 American National Standards Institute, "Fiber Channel Arbitrated Loop (FC-AL-2)", *American National Standard for Information Technology*, 1999, pp. i-v & 1-16.  
 American National Standards Institute, "Fiber Channel- Framing and Signaling (FC-FS)", *American National Standard for Information Technology*, Revision 1.80, Apr. 9, 2003, pp. i-xlvi & 1-18.  
 Deyring, K-P., "Serial ATA: High Speed Serialized AT Attachment", Jan. 7, 2003, Revision 1.0a, pp. 1-22.  
 Elliot, R.C. (Ed.), "Information Technology- SCSI Enclosure Services-2 (SES-2)", *Project T10/1559-D*, Jul. 15, 2003, Revision 4, pp. i-xiv & 1-79.  
 Elliot, R.C. (Ed.), "Information Technology- Serial Attached SCSI (SAS)", *Project T10/1562-D*, Jul. 9, 2003, Revision 5, pp. i-xxxii & 1-432.  
 Elliot, R., "Serial Attached SCSI Architecture", Sep. 2003, pp. 1-46.  
 Fairchild, S., "Expander Configuration Details", *T10 Document 02-359*, Sep. 9, 2002, pp. 1-17.  
 Fairchild, S., "SAS Expander Initiator Based Configuration", Jul. 12, 2002, pp. 1-32.  
 IEEE Computer Society, "Part 3: Carrier Sense Multiple Access With Collision Detection (CSMA/CS) Access Method and Physical Layer Specification", *IEEE Standard for Information Technology 802.3*, 2002, pp. 1-32.  
 McLean, P.T. (Ed.), "Information Technology- AT Attachment with Packet Interface-6 (ATA/ATAPI-6)", T13/1410D, Feb. 26, 2002, Revision 3B, pp. 1-56.

PCI Special Interest Group, "PCI Local Bus Specification", Mar. 29, 2002, Revision 2.3, pp. i-xiv & 1-6.  
 Phillips Semiconductors, "The I<sup>2</sup>C-Bus Specification", Jan. 200, Version 2.1, pp. 1-46.  
 Qlogic Corporation, "Enclosure Management Controller for Serial ATA", *Gem424 Data Sheet*, 2001, pp. 1-4.  
 U.S. Appl. No. 10/741,460, filed Dec. 18, 2003, entitled "Multiple Interfaces in a Storage Enclosure", invented by P.L. Seto.  
 U.S. Appl. No. 10/742,030, filed Dec. 18, 2003, entitled "Enclosure Management Device", invented by P.L. Seto.  
 U.S. Appl. No. 10/742,302, filed Dec. 18, 2003, entitled "Address Assignment for Adaptor Interfaces", invented by P.L. Seto.  
 U.S. Appl. No. 10/746,657, filed Dec. 23, 2003, entitled "Handling Redundant Paths Among Devices", invented by N.E. Marushak.  
 U.S. Appl. No. 10/787,074, filed Feb. 19, 2004, entitled "Generating Topology Information Identifying Devices in a Network Topology", invented by M.C. Clayton.  
 Siegl, C., and R.C. Klem, "Extension of PCI Bus to 21 Slots Using GTLP Transceiver", (Abstract Only), Proceedings of the IEEE Southeastcon 2000, Apr. 2000, pp. 326-331.  
 Technical Committee on Microprocessors and Microcomputers of the IEEE Computer Society, "IEEE Standard for Versatile Backplane Bus: VMEbus", (Abstract Only), ANSI/IEEE Std 1014-1987, 1987, 2 pp.  
 U.S. Appl. No. 11/551,218, filed Oct. 19, 2006, entitled "Multiple Interfaces in a Storage Enclosure", invented by P. Seto, 29 pp.  
 US Final Office Action, mailed May 8, 2006, for U.S. Appl. No. 10/741,460, 10 pp.  
 US Office Action, mailed Dec. 30, 2005, for U.S. Appl. No. 10/741,460, 14 pp.  
 U.S. Office Action, mailed May 10, 2007, for U.S. Appl. No. 10/742,030, 19 pp.  
 US Office Action, mailed May 17, 2007, for U.S. Appl. No. 11/551,215, 10 pp.  
 US Office Action, mailed Jul. 25, 2007, for U.S. Appl. No. 11/551,218, 10 pp.

\* cited by examiner

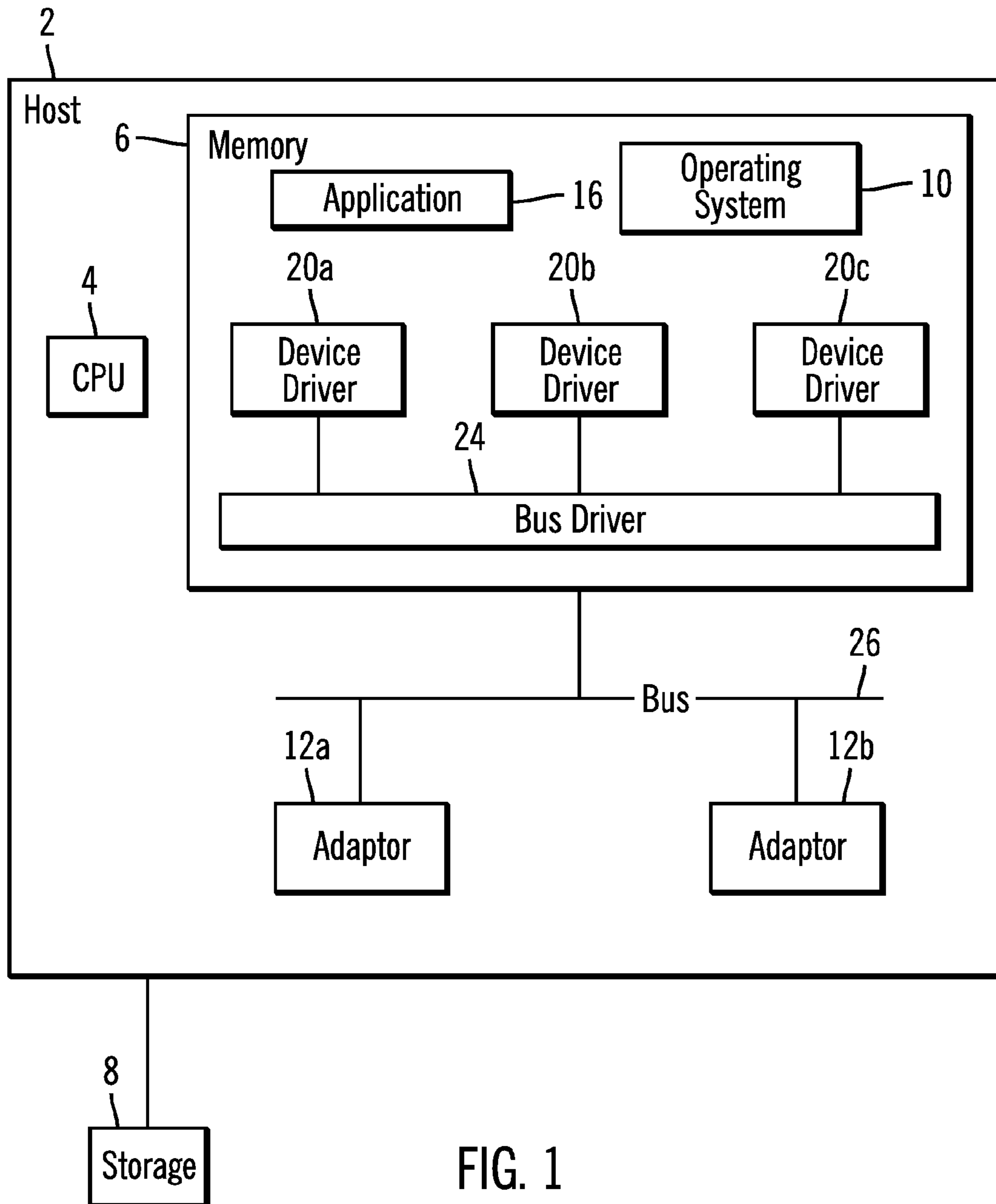


FIG. 1

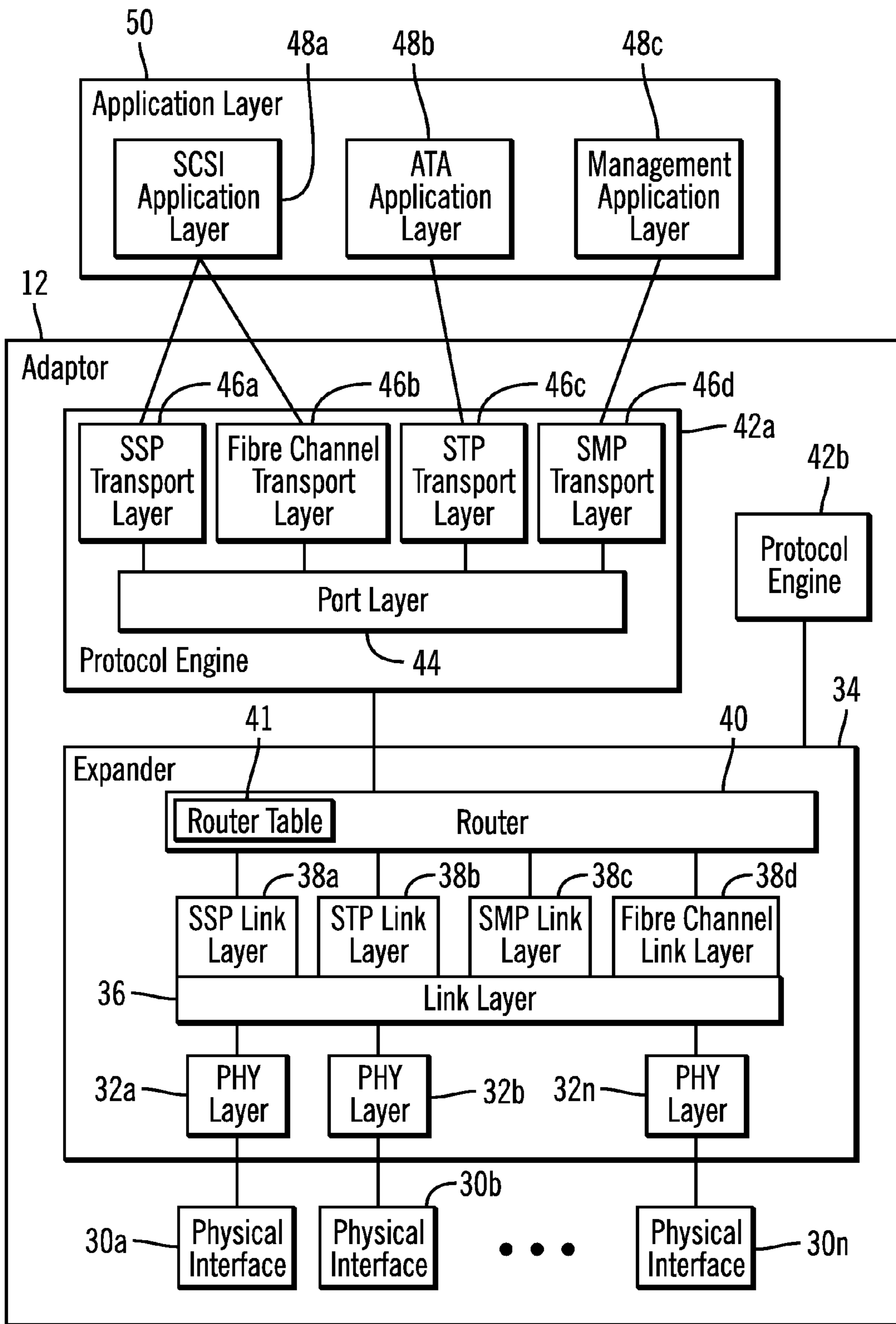


FIG. 2

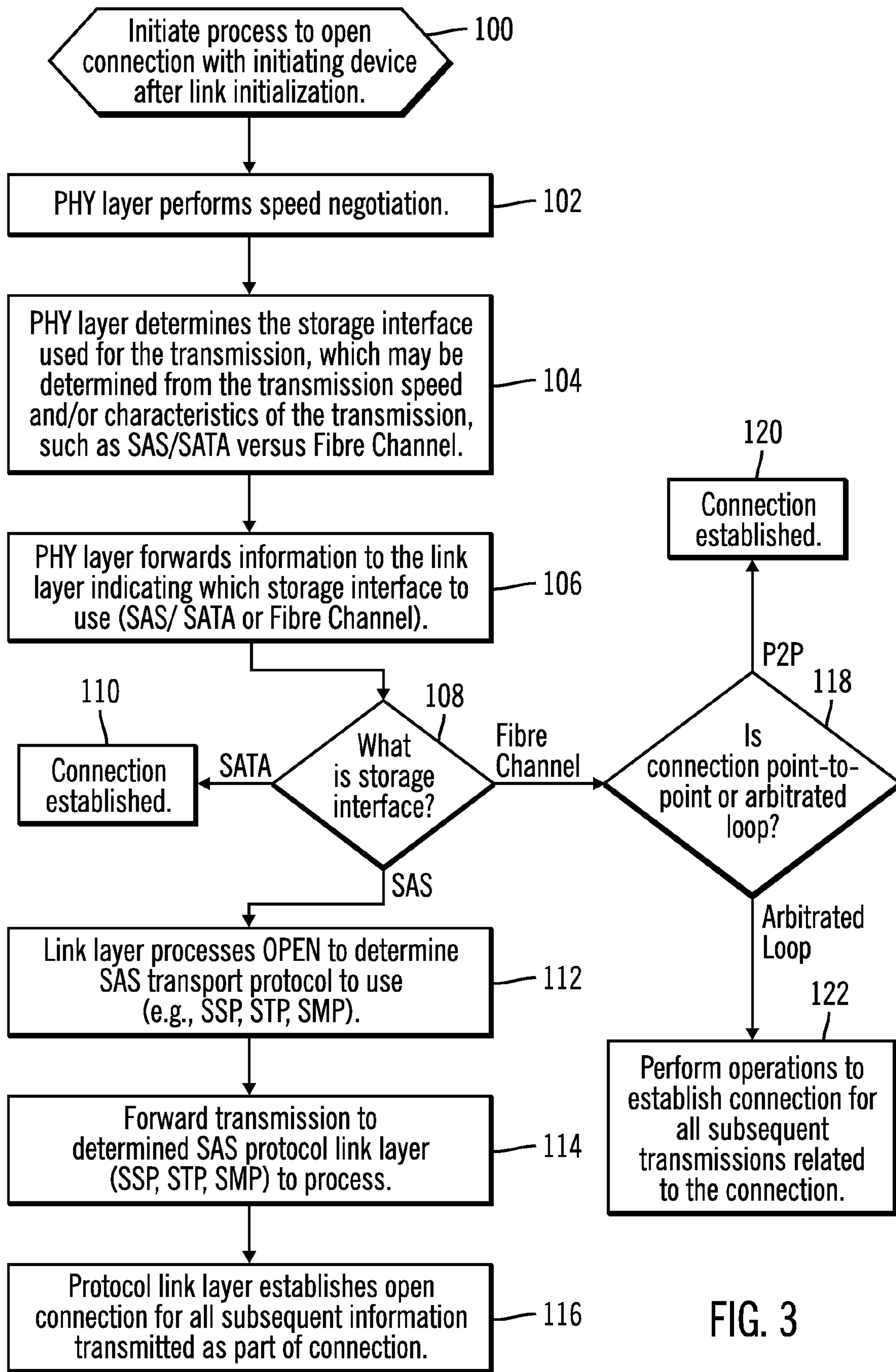


FIG. 3

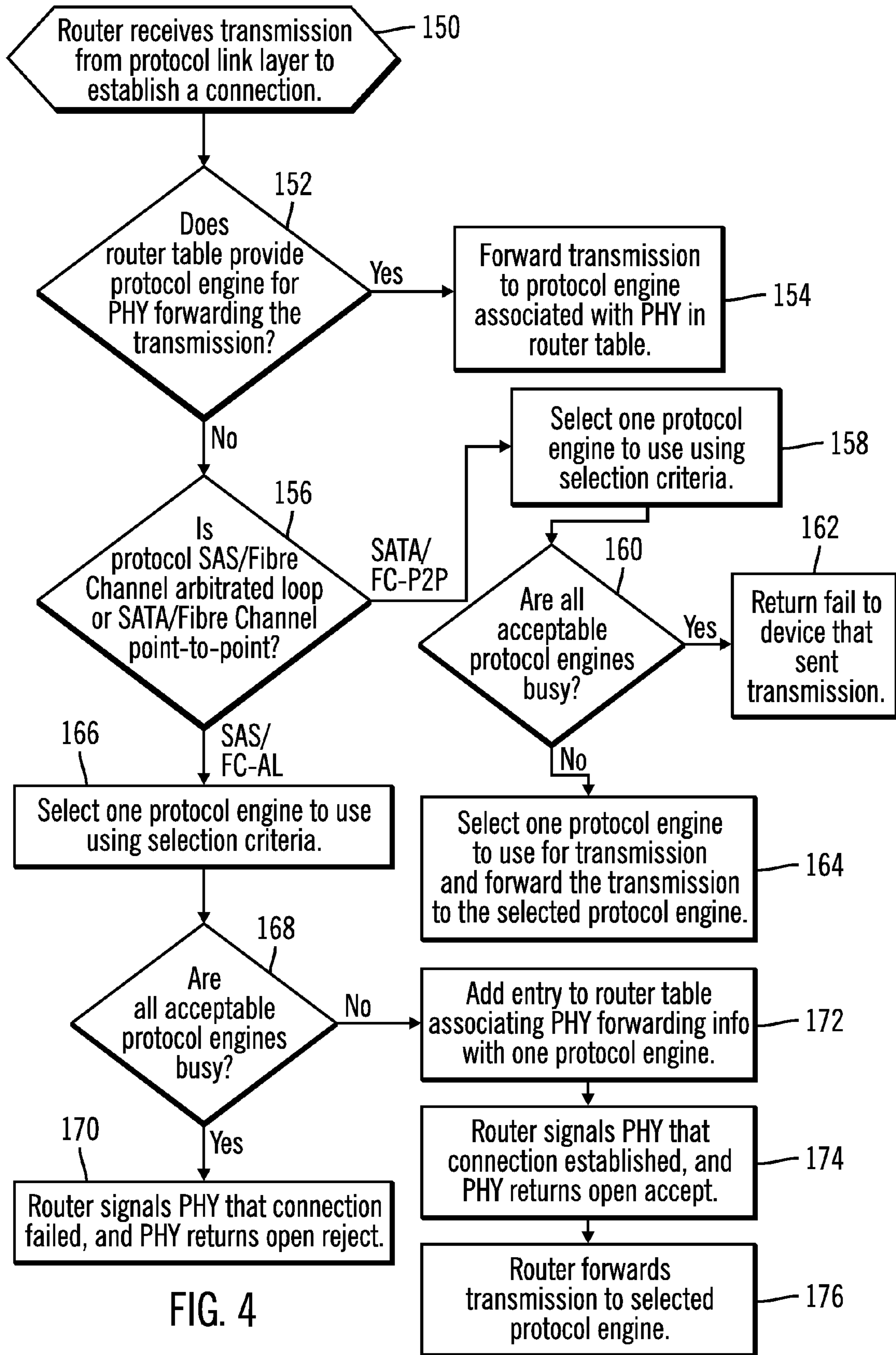


FIG. 4

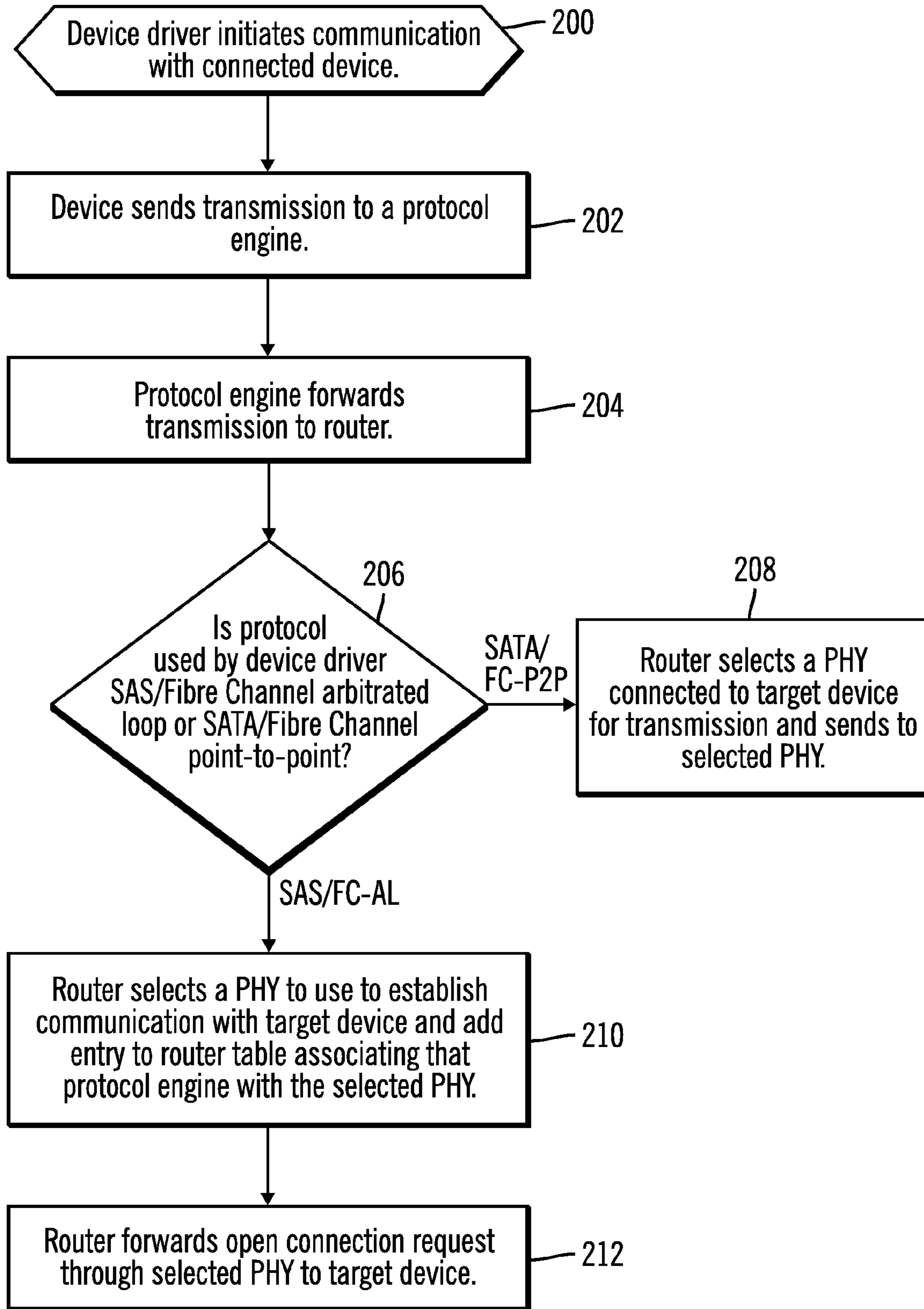


FIG. 5

## ADAPTOR SUPPORTING DIFFERENT PROTOCOLS

### CROSS REFERENCE TO RELATED APPLICATIONS

This application is related to the following and commonly assigned patent applications filed on the same date hereof:

“Multiple Interfaces In A Storage Enclosure”, by Pak-Lung Seto, having U.S. patent application Ser. No. 10/741,460; and

“Enclosure Management Device”, by Pak-Lung Set, having U.S. patent application Ser. No. 10/742,030.

### BACKGROUND

#### 1. Field

The present embodiments relate to a method, system, and program for supporting different storage interconnect architectures and transport protocols at an adaptor.

#### 2. Description of the Related Art

An adaptor or multi-channel protocol controller enables a device coupled to the adaptor to communicate with one or more connected end devices according to a storage interconnect architecture, also known as a hardware interface, where a storage interconnect architecture defines a standard way to communicate and recognize such communications, such as Serial Attached Small Computer System Interface (SCSI) (SAS), Serial Advanced Technology Attachment (SATA), Fibre Channel, etc. These storage interconnect architectures allow a device to maintain one or more connections to another end device via a point-to-point connection, an arbitrated loop of devices, an expander providing a connection to further end devices, or a fabric comprising interconnected switches providing connections to multiple end devices. In the SAS/SATA architecture, a SAS port is comprised of one or more SAS PHYs, where each SAS PHY interfaces a physical layer, i.e., the physical interface or connection, and a SAS link layer having multiple protocol link layer. Communications from the SAS PHYs in a port are processed by the transport layers for that port. There is one transport layer for each SAS port to interface with each type of application layer supported by the port. A “PHY” as defined in the SAS protocol is a device object that is used to interface to other devices and a physical interface. Further details on the SAS architecture for devices and expanders is described in the technology specification “Information Technology—Serial Attached SCSI (SAS)”, reference no. ISO/IEC 14776-150:200x and ANSI INCITS.\*\*\*:200x PHY layer (Jul. 9, 2003), published by ANSI; details on the Fibre Channel architecture are described in the technology specification “Fibre Channel Framing and Signaling Interface”, document no. ISO/IEC AWI 14165-25; details on the SATA architecture are described in the technology specification “Serial ATA: High Speed Serialized AT Attachment” Rev. 1.0A (January 2003).

Within an adaptor, the PHY layer performs the serial to parallel conversion of data, so that parallel data is transmitted to layers above the PHY layer, and serial data is transmitted from the PHY layer through the physical interface to the PHY layer of a receiving device. In the SAS specification, there is one set of link layers for each SAS PHY layer, so that effectively each link layer protocol engine is coupled to a parallel-to-serial converter in the PHY layer. A connection path connects to a port coupled to each PHY layer in the adaptor and terminate in a physical interface

within another device or on an expander device, where the connection path may comprise a cable or etched paths on a printed circuit board.

An expander is a device that facilitates communication and provides for routing among multiple SAS devices, where multiple SAS devices and additional expanders connect to the ports on the expander, where each port has one or more SAS PHYs and corresponding physical interfaces. The expander also extends the distance of the connection between SAS devices. The expander may route information from a device connecting to a SAS PHY on the expander to another SAS device connecting to the expander PHYs. In SAS, using the expander requires additional serial to parallel conversions in the PHY layers of the expander ports. Upon receiving a frame, a serial-to-parallel converter, which may be part of the PHY, converts the received data from serial to parallel to route internally to an output SAS PHY, which converts the frame from parallel to serial to the target device. The SAS PHY may convert parallel data to serial data through one or more encoders and convert serial data to parallel data through a parallel data builder and one or more decoders. A phased lock loop (PLL) may be used to track incoming serial data and lock into the frequency and phase of the signal. This tracking of the signal may introduce noise and error into the signal.

### BRIEF DESCRIPTION OF THE DRAWINGS

Referring now to the drawings in which like reference numbers represent corresponding parts throughout:

FIGS. 1 and 2 illustrate a system and adaptor in accordance with embodiments; and

FIGS. 3, 4, and 5 illustrate operations performed by the adaptor of FIGS. 1 and 2 to process frames in accordance with embodiments.

### DETAILED DESCRIPTION

In the following description, reference is made to the accompanying drawings which form a part hereof and which illustrate several embodiments. It is understood that other embodiments may be utilized and structural and operational changes may be made.

FIG. 1 illustrates a computing environment in which embodiments may be implemented. A host system 2 includes one or more central processing units (CPU) 4 (only one is shown), a volatile memory 6, non-volatile storage 8, an operating system 10, and one or more adaptors 12a, 12b which maintains physical interfaces to connect with other end devices directly in a point-to-point connection or indirectly through one or more expanders, one or more switches in a fabric or one or more devices in an arbitrated loop. An application program 16 further executes in memory 6 and is capable of transmitting to and receiving information from the target device through one of the physical interfaces in the adaptors 12a, 12b. The host 2 may comprise any computing device known in the art, such as a mainframe, server, personal computer, workstation, laptop, handheld computer, telephony device, network appliance, virtualization device, storage controller, etc. Various CPUs 4 and operating system 10 known in the art may be used. Programs and data in memory 6 may be swapped into storage 8 as part of memory management operations.

The operating system 10 may load a device driver 20a, 20b, 20c for each protocol supported in the adaptor 12a, 12b to enable communication with a device communicating using the supported protocol and also load a bus driver 24,



such as a Peripheral Component Interconnect (PCI) interface, to enable communication with a bus 26. Further details of PCI interface are described in the publication "PCI Local Bus, Rev. 2.3", published by the PCI-SIG. The operating system 10 may load device drivers 20a, 20b, 20c supported by the adaptors 12a, 12b upon detecting the presence of the adaptors 12a, 12b, which may occur during initialization or dynamically, such as the case with plug-and-play device initialization. In the embodiment of FIG. 1, the operating system 10 loads three protocol device drivers 20a, 20b, 20c. For instance, the device drivers 20a, 20b, 20c may support the SAS, SATA, and Fibre Channel point-to-point storage interfaces, i.e., interconnect architectures. Additional or fewer device drivers may be loaded based on the number of device drivers the adaptor 12 supports. FIG. 2 illustrates an embodiment of adaptor 12, which may comprise the adaptors 12a, 12b. Each adaptor includes a plurality of physical interfaces 30a, 30b . . . 30n, which may include the transmitter and receiver circuitry and other connection hardware. The physical interface may connect to another device via cables or a path etched on a printed circuit board so that devices on the printed circuit board communicate via etched paths. The physical interfaces 30a, 30b . . . 30n may provide different physical interfaces for different device connections, such as one physical interface 30a, 30b . . . 30n for connecting to a SAS/SATA device and another interface for a Fibre Channel device. Each physical interface 30a, 30b . . . 30n may be coupled to a PHY layer 32a, 32b . . . 32n within expander 34. The PHY layer 32a, 32b . . . 32n provides for an encoding scheme, such as 8b10b, to translate bits, and a clocking mechanism, such as a phased lock loop (PLL). The PHY layer 32a, 32b . . . 32n would include a serial-to-parallel converter to perform the serial-to-parallel conversion and the PLL to track the incoming data and provide the data clock of the incoming data to the serial-to-parallel converter to use when performing the conversion. Data is received at the adaptor 12 in a serial format, and is converted at the SAS PHY layer 32a, 32b . . . 32n to the parallel format for transmission within the adaptor 12. The SAS PHY layer 32a, 32b . . . 32n further provides for error detection, bit shift and amplitude reduction, and the out-of-band (OOB) signaling to establish an operational link with another SAS PHY in another device. The term interface may refer to the physical interface or the interface performing operations on the received data implemented as circuitry, or both.

The PHY layer 32a, 32b . . . 32n further performs the speed negotiation with the PHY in the external device transmitting data to adaptor 12. In certain embodiments, the PHY layer 32a, 32b . . . 32n may be programmed to allow speed negotiation and detection of different protocols transmitting at the same or different transmission speeds. For instance, SATA and SAS transmissions can be detected because they are transmitted at speeds of 1.5 gigahertz (GHz) and 3 GHz and Fibre Channel transmissions can be detected because they are transmitted at 1.0625 GHz, 2.125 GHz, and 4.25 GHz. Because link transmission speeds may be different for certain storage interfaces, the PHY layer 32a, 32b . . . 32n may detect storage interfaces having different link speeds by maintaining information on speeds for different storage interfaces. However, certain different storage interfaces, such as SAS and SATA, may transmit at the same link speeds and support common transport protocols. If storage interfaces transmit at a same link speed, then the PHY layer 32a, 32b . . . 32n may distinguish among storage interfaces capable of transmitting at the same speed by checking the transmission format to determine the storage

interface and protocol, where the link protocol defines the characteristics of the transmission, including speed and transmission data format.

For instance, the SAS and SATA protocol can be distinguished not only by their transmission speeds, but also by their use of the OOB signal. Other protocols, such as Fibre Channel do not use the OOB signal. Fibre Channel, SAS and SATA all have a four byte primitive. The primitive of SATA can be distinguished because the first byte of the SATA primitive indicates "K28.3", whereas the first byte of the SAS and Fibre Channel primitive indicates "K28.5". The SAS and Fibre Channel primitives can be distinguished based on the content of the next three bytes of their primitives, which differ. Thus, the content of the primitives can be used to distinguish between the SAS, SATA and Fibre Channel protocols. Additionally, different of the protocols, such as SAS and Fibre Channel have different handshaking protocols. Thus, the handshaking protocol being used by the device transmitting the information can be used to distinguish the storage connect interface being used.

The PHY layer 32a, 32b . . . 32n forwards the frame to the link layer 36 in the expander 34. The link layer 36 may maintain a set of elements for each protocol supported by a port, such as a Serial SCSI Protocol (SSP) link layer 38 to process SSP frames, a Serial Tunneling Protocol (STP) layer 38b, a Serial Management Protocol (SMP) layer 38c, and a Fibre Channel link layer 38d to support the Fibre Channel protocol for transporting the frames. Within the expander 34, information is routed from one PHY to another. The transmitted information may include primitives, packets, frames, etc., and may be used to establish the connection and open the address frame. A router 40 routes transmissions between the protocol engines 42a, 42b and the PHY layers 32a, 32b . . . 32n. The router 40 maintains a router table 41 providing an association of PHY layers 32a, 32b . . . 32n to protocol engines 42a, 42b, such that a transmission from a PHY layer or protocol engine is routed to the corresponding protocol engine or PHY layer, respectively, indicated in the router table 41. If the protocol engines 42a, 42b support the transport protocol, e.g., SSP, STP, SMP, Fibre Channel protocol, etc., associated with the link layer 38a, 38b, 38c, 38d forwarding the transmission, then the router 40 may use any technique known in the art to select among the multiple protocol engines 42a, 42b to process the transmission, such as round robin, load balancing based on protocol engine 42a, 42b utilization, etc. The Fibre Channel Protocol comprises the transport layer for handling information transmitted on a Fibre Channel storage interface. Data may be communicated in frames, packets, primitives or any other data transmission format known in the art. A transport layer comprises any circuitry, including software or hardware, that is use to provide a virtual error-free, point to point connection to allow for the transmission of information between devices so that transmitted information arrives un-corrupted and in the correct order. The transport layer further establishes, e.g., opens, and dissolves connections between devices.

A transport protocol provides a set of transmission rules and handshaking procedures used to implement a transport layer, often defined by an industry standard, such as SAS, SATA, Fibre Channel, etc. The transport layer and protocol may comprise those transport protocols described herein and others known in the art. The protocol engine 42a, 42b comprises the hardware and/or software that implements different transport protocols to provide transport layer functionality for different protocols.

Each protocol engine 42a, 42b is capable of performing protocol related operations for all the protocols supported by

the adaptor **12**. Alternatively, different protocol engines may support different protocols. For instance, protocol engine **42b** may support the same transport layers as protocol engine **42a** or a different set of transport layers. Each protocol engine **42a, 42b** implements a port layer **44**, and a transport layer, such as a SSP transport layer **46a**, STP transport layer **46b**, SMP transport layer **46c**, and a Fibre Channel Protocol transport layer **46d**. Further, the protocol engines **30a, 30b** may support the transport and network layer related operations for the supported protocols. The port layer **44** interfaces between the link layers **38a, 38b, 38c, 38d** via the router **40** and the transport layers **46a, 46b, 46c, 46d** to transmit information to the correct transport layer or link layer. The PHYs **32a, 32b . . . 32n** and corresponding physical interfaces **30a, 30b . . . 30n** may be organized into one or more ports, where each SAS port has a unique SAS address. The port comprises a component or construct to which interfaces are assigned. An address comprises any identifier used to identify a device or component. The protocol engines **42a, 42b** may further include one or more virtual PHY layers to enable communication with virtual PHY layers in the router **40**. A virtual PHY is an internal PHY that connects to another PHY inside of the device, and not to an external PHY. Data transmitted to the virtual PHY typically does not need to go through a serial-to-parallel conversion.

Each protocol engine **42a, 42b** includes an instance of the protocol transport layers **46a, 46b, 46c, 46d**, where there is one transport layer to interface with each type of application layer **48a, 48b, 48c** in the application layer **50**. The application layer **50** may be supported in the adaptor **12** or host system **2** and provides network services to the end users. For instance, the SSP transport layer **46a** and Fibre Channel Protocol (FCP) transport layer **46b** interface with a SCSI application layer **48a**, the STP transport layer **46c** interfaces with an Advanced Technology Attachment (ATA) application layer **48b**, and the SMP transport layer **46d** interfaces with a management application layer **48c**. Further details of the ATA technology are described in the publication "Information Technology—AT Attachment with Packet Interface—6 (ATA/ATAPI-6)", reference no. ANSI INCITS 361-2002 (September, 2002).

All the PHY layers **32a, 32b . . . 32n** may share the same link layer and protocol link layers, or there may be a separate instance of each link layer and link layer protocol **38a, 38b, 38c, 38d** for each PHY. Further, each protocol engine **42a, 42b** may include one port layer **44** for all ports including the PHY layers **32a, 32b . . . 32n** or may include a separate instance of the port layer **44** for each port in which one or more PHY layers and the corresponding physical interfaces are organized. Further details on the operations of the physical layer, PHY layer, link layer, port layer, transport layer, and application layer and components implementing such layers described herein are found in the technology specification "Information Technology—Serial Attached SCSI (SAS)", referenced above.

The router **40** allows the protocol engines **42a, 42b** to communicate to any of the PHY layers **32a, 32b . . . 32n**. The protocol engines **42a, 42b** communicate parallel data to the PHY layers **32a, 32b . . . 32n**, which include parallel-to-serial converters to convert the parallel data to serial data for transmittal through the corresponding physical interface **30a, 30b . . . 30n**. The data may be communicated to a PHY on the target device or an intervening external expander. A target device is a device to which information is transmitted from a source or initiator device attempting to communicate with the target device.

With the described embodiments of FIGS. **1** and **2**, one protocol engine **42a, 42b** having the port and transport layers can manage transmissions to multiple PHY layers **32a, 32b . . . 32n**. The transport layers **46a, 46b, 46c, 46d** of the protocol engines **42a, 42b** may only engage with one open connection at a time. However, if delays are experienced from the target on one open connection, the protocol engine **42a, 42b** can disconnect and establish another connect to process I/O requests from that other connection to avoid latency delays for those target devices trying to establish a connection. This embodiment provides greater utilization of the protocol engine bandwidth by allowing each protocol engine to multiplex among multiple target devices and switch among connections. The protocol engines **42a, 42b** and physical interface have greater bandwidth than the target device, so that the target device throughput is lower than the protocol engine **42a, 42b** throughput. In certain embodiments, the protocol engines **42a, 42b** may multiplex between different PHYs **32a, 32b . . . 32n** to manage multiple targets.

Allowing one protocol engine to handle multiple targets further reduces the number of protocol engines that need to be implemented in the adaptor to support all the targets.

FIG. **3** illustrates operations performed by the PHY layers **32a, 32b . . . 32n** and the link layer **36** to open a connection with an initiating device, where the initiating device may transmit using SAS, Fibre Channel, or some other storage interface (storage interconnect architecture). The operation to establish the connection may occur after the devices are discovered during identification and link initialization. In response to a reset or power-on sequence, the PHY layer **32a, 32b** may begin (at block **100**) link initialization by receiving link initialization information, such as primitives, from an initiator device at one physical interface **30a, 30b . . . 30n** (FIG. **2**). The PHY layer **32a, 32b . . . 32n** coupled to the receiving physical interface **30a, 30b . . . 30n** performs (at block **102**) speed negotiation to ensure that the link operates at the highest frequency. In certain embodiments, the PHY layer **32a, 32b . . . 32n** includes the capability to detect and negotiate speeds for different storage interfaces, where the different storage interfaces have different transmission characteristics, such as different transmission speeds and/or transmission information, such as is the case with the SAS/SATA and Fibre Channel storage interfaces. The PHY layer **32a, 32b . . . 32n** then determines (at block **104**) the storage interface used for the transmission to establish the connection, which may be determined from the transmission speed if a unique transmission speed is associated with a storage interface or from characteristics of the transmission, such as information in the header of the transmission, format of the transmission, etc. The PHY layer **32a, 32b** forwards (at block **106**) the information to the link layer **36** indicating which detected storage interface to use (SAS/SATA or Fibre Channel).

If (at block **108**) the determined storage interface complies with the SATA protocol, then the connection is established (at block **110**) and no further action is necessary. If (at block **108**) the connection utilizes the SAS protocol, then the link layer **36** processes (at block **112**) an OPEN frame to determine the SAS transport protocol to use (e.g., SSP, STP, SMP, Fibre Channel Protocol). The OPEN frame is then forwarded (at block **114**) to the determined SAS protocol link layer **38a, 38b, 38c, 38d** (SSP, STP, SMP, Fibre Channel Protocol) to process. The protocol link layer **38a, 38b, 38c, 38d** then establishes (at block **116**) an open connection for all subsequent frames transmitted as part of that opened connection. The connection must be opened using the OPEN frame between an initiator and target port before commu-

nication may begin. A connection is established between one SAS initiator PHY in the SAS initiator port and one SAS target PHY in the SAS target port. If (at blocks **108** and **118**) the storage interface complies with a point-to-point Fibre Channel protocol, then the connection is established (at block **120**). Otherwise, if (at blocks **108** and **118**) the storage interface complies with the Fibre Channel Arbitrated Loop protocol, then the Fibre Channel link layer **38d** establishes (at block **122**) the open connection for all subsequent frames transmitted as part of connection. The Fibre Channel link layer **38d** may establish the connection using Fibre Channel open primitives. Further details of the Fibre Channel Arbitrated Loop protocol are described in the publication “Information Technology—Fibre Channel Arbitrated Loop (FC-AL-2)”, having document no. ANSI INCITS 332-1999.

With the described implementations, the PHY layer **32a**, **32b** . . . **32n** is able to determine the storage interface for different storage interfaces that transmit at different transmission link speeds and/or have different transmission characteristics. This determined storage interface information is then forwarded to the link layer **36** to use to determine which link layer protocol and transport protocol to use to establish the connection, such as a SAS link layer protocol, e.g., **38a**, **38b**, **38c**, or the Fibre Channel link layer protocol **38d**, where the different protocols that may be used require different processing to handle.

FIG. **4** illustrates operations performed by the router **40** to select a protocol engine **42a**, **42b** to process the received frame. Upon receiving (at block **150**) a transmission from the protocol link layer **38a**, **38b**, **38c**, **38d**, such as a frame, packet, primitive, etc., to establish a connection, if (at block **152**) a router table **41** provides an association of a protocol engine **42a**, **42b** for the PHY **32a**, **32b** . . . **32n** forwarding the transmission, then the router **40** forwards (at block **154**) the transmission to the protocol engine **42a**, **42b** associated with the PHY indicated in the router table **41**. If (at block **152**) the router table **41** does not provide an association of a PHY layer and protocol engine and if (at block **156**) the protocol of the transmission complies with the SATA or Fibre Channel point-to-point protocol, then the router **40** selects (at block **158**) one protocol engine to use based on a selection criteria, such as load balancing, round robin, etc. If (at block **160**) all protocol engines **46a**, **46b** capable of handling the determined protocol are busy, then fail is returned (at block **162**) to the device that sent a transmission. Otherwise, if (at block **160**) a protocol engine **46a**, **46b** is available, then one protocol engine **46a**, **46b** is selected (at block **164**) to use for the transmission and the transmission is forwarded to the selected protocol engine.

If (at block **156**) the protocol of the connection request complies with the SAS or Fibre Channel Arbitrated Loop protocol, then the router **40** selects (at block **166**) one protocol engine **46a**, **46b** to use based on a selection criteria. If (at block **168**) all protocol engines **46a**, **46b** capable of handling the determined protocol are busy, then the PHY receiving the transmission is signaled that the connection request failed, and the PHY **32a**, **32b** . . . **32n** returns (at block **170**) an OPEN reject command to the transmitting device. Otherwise, if (at block **168**) a protocol engine **46a**, **46b** is available, then an entry is added (at block **172**) to the router table **41** associating the PHY **42a**, **42b** . . . **42n** forwarding the transmission with one protocol engine **46a**, **46b**. The router **40** signals (at block **174**) the PHY that the connection is established, and the PHY returns OPEN accept. The router **40** forwards (at block **176**) the transmission to the selected protocol engine **46a**, **46b**.

Additionally, the application layer **50** may open a connection to transmit information to a target device by communicating the open request frames to one protocol engine **42a**, **42b**, using load balancing or some other selecting technique, where the protocol engine **42a**, **42b** transport and port layers transmit the open connection frames to the router **40** to direct the link initialization to the appropriate link layer and PHY layer.

FIG. **5** illustrates operations performed in the adaptor **12** to enable a device driver **20a**, **20b**, **20c** to communicate information to a target device through an adaptor **12a**, **12b** (FIG. **1**). At block **200**, a device driver **20a**, **20b**, **20c** transmits information to initiate communication with a connected device by sending (at block **202**) information to a protocol engine **46a**, **46b**. A device driver **20a**, **20b**, **20c** may perform any operation to select a protocol engine to use. The protocol engine **46a**, **46b** receiving the transmission forwards (at block **204**) the transmission to the router **40**. If (at block **206**) the protocol used by the device driver **20a**, **20b**, **20c** is SATA or Fibre Channel point-to-point protocol, then the router **40** selects (at block **208**) a PHY **32a**, **32b** . . . **32n** connected to the target device (directly or indirectly through one or more expanders or a fabric) for transmission and sends the transmission to the selected PHY. If (at block **206**) the protocol used by the device driver **20a**, **20b**, **20c** initiating the transmission is SAS or Fibre Channel Arbitrated Loop, then the router **40** selects (at block **210**) a PHY **32a**, **32b** . . . **32n** to use to establish communication with the target device and add an entry to the router table associating the protocol engine **42a**, **42b** forwarding the transmission with the selected PHY, so that the indicated protocol engine and PHY are used for communications through that SAS or Fibre Channel Arbitrated Loop connection. The router **40** then forwards (at block **212**) the open connection request through the selected PHY **32a**, **32b** . . . **32n** to the target device.

Described embodiments provide techniques for allowing connections with different storage interfaces that communicate at different transmission speeds and/or different transmission characteristics. In this way, a single adaptor **12** may provide multiple connections for different storage interfaces (storage interconnect architectures) that communicate using different transmission characteristics, such as transmitting at different link speeds or including different protocol information in the transmissions. For instance, the adaptor **12** may be included in an enclosure that is connected to multiple storage devices on a rack or provides the connections for storage devices within the same enclosure.

Still further, with the described embodiments, there may be only one serial to parallel conversion between the PHY layers **32a**, **32b** . . . **32n** performing parallel-to-serial conversion and the protocol engines **42a**, **42b** within the adaptor. In implementations where the expander is located external to the adaptor, three parallel-to-serial conversions may be performed to communicate data from the connections to the router (serial to parallel), from the router in the expander to the adaptor (parallel to serial), and at the adaptor from the connection to the protocol engine (serial to parallel). Certain described embodiments eliminate the need for two of these conversions by allowing the parallel data to be transmitted directly from the router to the protocol engines in the same adaptor component. Reducing the number of parallel to serial conversions and corresponding PLL tracking reduces data and bit errors that may be introduced by the frequency changes produced by the PLL in the converters and may reduce latency delays caused by such additional conversions.

## ADDITIONAL EMBODIMENT DETAILS

The described embodiments may be implemented as a method, apparatus or article of manufacture using programming and/or engineering techniques to produce software, firmware, hardware, or any combination thereof. The term “article of manufacture” and “circuitry” as used herein refers to a state machine, code or logic implemented in hardware logic (e.g., an integrated circuit chip, Programmable Gate Array (PGA), Application Specific Integrated Circuit (ASIC), etc.) or a computer readable medium, such as magnetic storage medium (e.g., hard disk drives, floppy disks, tape, etc.), optical storage (CD-ROMs, optical disks, etc.), volatile and non-volatile memory devices (e.g., EEPROMs, ROMs, PROMs, RAMs, DRAMs, SRAMs, firmware, programmable logic, etc.). Code in the computer readable medium is accessed and executed by a processor. When the code or logic is executed by a processor, the circuitry would include the medium including the code or logic as well as the processor that executes the code loaded from the medium. The code in which preferred embodiments are implemented may further be accessible through a transmission media or from a file server over a network. In such cases, the article of manufacture in which the code is implemented may comprise a transmission media, such as a network transmission line, wireless transmission media, signals propagating through space, radio waves, infrared signals, etc. Thus, the “article of manufacture” may comprise the medium in which the code is embodied. Additionally, the “article of manufacture” may comprise a combination of hardware and software components in which the code is embodied, processed, and executed. Of course, those skilled in the art will recognize that many modifications may be made to this configuration, and that the article of manufacture may comprise any information bearing medium known in the art.

Additionally, the expander, PHYs, and protocol engines may be implemented in one or more integrated circuits on the adaptor or on the motherboard.

In the described embodiments, layers were shown as operating within specific components, such as the expander and protocol engines. In alternative implementations, layers may be implemented in a manner different than shown. For instance, the link layer and link layer protocols may be implemented with the protocol engines or the port layer may be implemented in the expander.

In the described embodiments, the protocol engines each support multiple transport protocols. In alternative embodiments, the protocol engines may support different transport protocols, so the expander 40 would direct communications for a particular protocol to that protocol supporting the determined protocol.

In the described embodiments, transmitted information is received at an adaptor card from a remote device over a connection. In alternative embodiments, the transmitted and received information processed by the transport protocol layer or device driver may be received from a separate process executing in the same computer in which the device driver and transport protocol driver execute.

In certain implementations, the device driver and network adaptor embodiments may be included in a computer system including a storage controller, such as a SCSI, Redundant Array of Independent Disk (RAID), etc., controller, that manages access to a non-volatile or volatile storage device, such as a magnetic disk drive, tape media, optical disk, etc. In alternative implementations, the network adaptor embodi-

ments may be included in a system that does not include a storage controller, such as certain hubs and switches.

In certain implementations, the adaptor may be configured to transmit data across a cable connected to a port on the adaptor. In further embodiments, the adaptor may be configured to transmit data across etched paths on a printed circuit board. Alternatively, the adaptor embodiments may be configured to transmit data over a wireless network or connection.

In described embodiments, the storage interfaces supported by the adaptors comprised SATA, SAS and Fibre Channel. In additional embodiments, other storage interfaces may be supported. Additionally, the adaptor was described as supporting certain transport protocols, e.g. SSP, Fibre Channel Protocol, STP, and SMP. In further implementations, the adaptor may support additional transport protocols used for transmissions with the supported storage interfaces. The supported storage interfaces may transmit using different transmission characteristics, e.g., different link speeds and different protocol information included with the transmission. Further, the physical interfaces may have different physical configurations, i.e., the arrangement and number of pins and other physical interconnectors, when the different supported storage interconnect architectures use different physical configurations.

The adaptor 12 may be implemented on a network card, such as a Peripheral Component Interconnect (PCI) card or some other I/O card, or on integrated circuit components mounted on a system motherboard or backplane.

The illustrated logic of FIGS. 3, 4, and 5 show certain events occurring in a certain order. In alternative embodiments, certain operations may be performed in a different order, modified or removed. Moreover, operations may be added to the above described logic and still conform to the described embodiments. Further, operations described herein may occur sequentially or certain operations may be processed in parallel. Yet further, operations may be performed by a single processing unit or by distributed processing units.

The foregoing description of various embodiments has been presented for the purposes of illustration and description. It is not intended to be exhaustive or to limit the embodiments to the precise form disclosed. Many modifications and variations are possible in light of the above teaching.

What is claimed is:

1. A method, comprising:

receiving a transmission at one of multiple connections; maintaining information on storage interconnect architectures and transmission characteristics, wherein the storage interconnect architectures have different transmission characteristics;

determining at least one transmission characteristic of the received transmission;

determining from the information the storage interconnect architecture associated with the determined transmission characteristic;

using the information on the determined storage interconnect architecture to process the transmission and determine one of a plurality of available transport layers for the received transmission, wherein there is one transport layer for each supported transport protocol; and

## 11

forwarding the transmission to the determined transport layer by:

selecting one of a plurality of protocol engines by performing load balancing among the available protocol engines to select a least utilized protocol engine; and

forwarding the transmission to the selected protocol engine, wherein the determined transport layer in the selected protocol engine processes the transmission.

2. The method of claim 1, wherein each protocol engine is capable of processing transmissions from different devices using different storage interconnect architectures.

3. The method of claim 2, wherein a port comprises one or more connections through one or more physical interfaces, and wherein each protocol engine and the transport layers therein are capable of processing transmissions from multiple ports.

4. The method of claim 3, wherein a link layer uses the information on the determined storage interconnect architecture to process the transmission and determine a transport layer, and wherein each protocol engine includes a port layer to interface between the link layer and transport layers in the protocol engine.

5. The method of claim 1, wherein each protocol engine supports a same set of transport layers and transport layers used by different storage interconnect architectures.

6. The method of claim 1, wherein the received transmission comprises a connection open frame, and wherein a connection is established using the determined transport protocol and selected protocol engine for all frames communicated over that open connection until the connection is closed.

7. The method of claim 1, further comprising:

forwarding the transmission to the protocol engine in response to determining a first storage interconnect architecture; and

forwarding the transmission to a protocol link layer to open a connection in response to determining a second storage interconnect architecture.

8. The method of claim 7, further comprising:

associating the selected protocol engine with the connection in response to determining the second storage interconnect architecture.

9. The method of claim 1, wherein the supported storage interconnect architectures comprise SATA, SAS, and Fibre Channel and wherein the supported transport protocols comprise SSP, Fibre Channel Protocol, STP, and SMP.

10. The method of claim 1, wherein the transmission characteristics include a link speed, wherein different storage interconnect architectures transmit at different link speeds, and wherein the link speed is used to distinguish between storage interconnect architectures.

11. The method of claim 10, wherein the transmission characteristics include characteristics of a data format of the transmission, wherein different storage interconnect architectures transmit using different data formats, and wherein the data format is used to distinguish between storage interconnect architectures that transmit at same link speeds.

12. An adaptor, comprising:

a plurality of physical interfaces capable of interfacing with multiple storage interconnect architectures;

transport layers, wherein there is one transport layer for each supported transport protocol;

a plurality of protocol engines each supporting multiple of the transport layers;

information on storage interconnect architectures and transmission characteristics used for the storage inter-

## 12

connect architectures, wherein the storage interconnect architectures have different transmission characteristics;

circuitry capable of causing operations to be performed, the operations comprising:

receiving a transmission at one of the physical interfaces;

determining at least one transmission characteristic of the received transmission;

determining from the information the storage interconnect architecture associated with the determined transmission characteristic;

using the information on the determined storage interconnect architecture to process the transmission and determine one of a plurality of available transport layers for the received transmission, wherein there is one transport layer for each supported transport protocol; and

forwarding the transmission to the determined transport layer by;

selecting one of the plurality of protocol engines by performing load balancing among the available protocol engines to select a least utilized protocol engine; and forwarding the transmission to the selected protocol engine, wherein the determined transport layer in the selected protocol engine processes the transmission.

13. The adaptor of claim 12, wherein at least one physical interface has a first physical configuration to interface with a first storage interconnect architecture and at least one physical interface has a second physical configuration to interface with a second interconnect architecture.

14. The adaptor of claim 12, wherein each protocol engine is capable of processing frames transmitted on different connections through different physical interfaces.

15. The adaptor of claim 12, further comprising:

a plurality of ports, wherein each port comprises one or more physical interfaces, and wherein each protocol engine and the transport layers therein are capable of processing frames from multiple ports.

16. The adaptor of claim 15, further comprising:

a link layer using the information on the determined storage interconnect architecture to process the frame and determine a transport layer; and

a port layer in each protocol engine to interface between the link layer and transport layers in the protocol engine.

17. The adaptor of claim 12, wherein each protocol engine supports a same set of transport layers and transport layers used by different storage interconnect architectures.

18. The adaptor of claim 12, wherein the received transmission comprises a connection open frame, and wherein a connection is established using the determined transport protocol and selected protocol engine for all frames communicated over that open connection until the connection is closed.

19. The adaptor of claim 12, wherein the circuitry is further capable of performing:

forwarding the transmission to the protocol engine in response to determining a first storage interconnect architecture; and

forwarding the transmission to a protocol link layer to open a connection in response to determining a second storage interconnect architecture.

## 13

20. The adaptor of claim 19, wherein the circuitry is further capable of performing:

associating the selected protocol engine with the connection in response to determining the second storage interconnect architecture.

21. The adaptor of claim 13, wherein the supported storage interconnect architectures comprise SATA, SAS, and Fibre Channel and wherein the supported transport protocols comprise SSP, Fibre Channel Protocol, STP, and SMP.

22. The adaptor of claim 12, further comprising:

a PHY layer coupled to each physical interface, wherein the PHY layer determines the transmission characteristics of the received transmission and determines from the information the storage interconnect architecture associated with a determined link speed.

23. A system, comprising:

a processor;

a plurality of protocol engines each supporting multiple transport layers; and

at least one adaptor in data communication with the processor, comprising:

plurality of physical interfaces capable of interfacing with multiple storage interconnect architectures including SAS, SATA, and Fibre Channel;

information on storage interconnect architectures and transmission characteristics, wherein the storage interconnect architectures have different transmission characteristics;

transport layers, wherein there is one transport layer for each supported transport protocol;

circuitry capable of causing operations to be performed, the operations comprising:

receiving a transmission at one of the physical interfaces;

determining at least one transmission characteristic of the received transmission;

determining from the information the storage interconnect architecture associated with the determined transmission characteristic;

using the information on the determined storage interconnect architecture to process the transmission and determine one of a plurality of available transport layers for the received transmission, wherein there is one transport layer for each supported transport protocol; and

forwarding the transmission to the determined transport layer by;

selecting one of the plurality or protocol engines by performing load balancing among the available protocol engines to select a least utilized protocol engine; and forwarding the transmission to the selected protocol engine, wherein the determined transport layer in the selected protocol engine processes the transmission.

24. The system of claim 23, wherein on at least one adaptor, at least one physical interface has a first physical configuration to interface with a first storage interconnect architecture and at least one physical interface has a second physical configuration to interface with a second interconnect architecture, wherein the first and second storage interconnect architectures communicate using different transmission characteristics.

25. An article of manufacture, wherein the article of manufacture causes operations to be performed, the operations comprising:

## 14

receiving a transmission at one of multiple connections; maintaining information on storage interconnect architectures and transmission characteristics, wherein the storage interconnect architectures have different transmission characteristics;

determining at least one transmission characteristic of the received transmission;

determining from the information the storage interconnect architecture associated with the determined transmission characteristic;

using the information on the determined storage interconnect architecture to process the transmission and determine one of a plurality of available transport layers for the received transmission, wherein there is one transport layer for each supported transport protocol; and forwarding the transmission to the determined transport layer by;

selecting one of the plurality or protocol engines by performing load balancing among the available protocol engines to select a least utilized protocol engine; and forwarding the transmission to the selected protocol engine, wherein the determined transport layer in the selected protocol engine processes the transmission.

26. The article of manufacture of claim 25, wherein each protocol engine is capable of processing transmissions from different connections using different storage interconnect architectures.

27. The article of manufacture of claim 26, wherein a port comprises one or more connections through one or more physical interfaces, and wherein each protocol engine and the transport layers therein are capable of processing transmissions from multiple ports.

28. The article of manufacture of claim 27, wherein a link layer uses the information on the determined storage interconnect architecture to process the transmission and determine a transport layer, and wherein each protocol engine includes a port layer to interface between the link layer and transport layers in the protocol engine.

29. The article of manufacture of claim 25, wherein each protocol engine supports a same set of transport layers and transport layers used by different storage interconnect architectures.

30. The article of manufacture of claim 25, wherein the received transmission comprises a connection open frame, and wherein a connection is established using the determined transport protocol and selected protocol engine for all frames communicated over that open connection until the connection is closed.

31. The article of manufacture of claim 25, wherein the operations further comprise:

forwarding the transmission to the protocol engine in response to determining a first storage interconnect architecture; and

forwarding the transmission to a protocol link layer to open a connection in response to determining a second storage interconnect architecture.

32. The article of manufacture of claim 31, wherein the operations further comprise:

associating the selected protocol engine with the connection in response to determining the second storage interconnect architecture.

33. The article of manufacture of claim 25, wherein the supported storage interconnect architectures comprise SATA, SAS, and Fibre Channel and wherein the supported transport protocols comprise SSP, Fibre Channel Protocol, STP, and SMP.

**15**

**34.** The article of manufacture of claim **25**, wherein the transmission characteristics include a link speed, wherein different storage interconnect architectures transmit at different link speeds, and wherein the link speed is used to distinguish between storage interconnect architectures.

**35.** The article of manufacture of claim **34**, wherein the transmission characteristics include characteristics of a data format of the transmission, wherein different storage interconnect architectures transmit using different data formats,

**16**

and wherein the data format is used to distinguish between storage interconnect architectures that transmit at same link speeds.

**36.** The article of manufacture of claim **25**, wherein the article of manufacture stores instructions that when executed result in performance of the operations.

\* \* \* \* \*

UNITED STATES PATENT AND TRADEMARK OFFICE  
**CERTIFICATE OF CORRECTION**

PATENT NO. : 7,376,147 B2  
APPLICATION NO. : 10/742029  
DATED : May 20, 2008  
INVENTOR(S) : Pak-Lung Seto and Deif N. Atallah

Page 1 of 1

It is certified that error appears in the above-identified patent and that said Letters Patent is hereby corrected as shown below:

Column 11, Line 3, claim 1 delete "plurality or protocol" and replace with --plurality of protocol--.

Column 12, Line 22, claim 12 delete "plurality or protocol" and replace with --plurality of protocol--.

Column 13, Line 50, claim 23 delete "plurality or protocol" and replace with --plurality of protocol--.

Column 14, Line 18, claim 25 delete "the plurality or protocol" and replace with --a plurality of protocol--.

Signed and Sealed this

Fifteenth Day of September, 2009



David J. Kappos  
*Director of the United States Patent and Trademark Office*